ショートリードマッピング高速計算システムの研究

著者	曽我部 陽光
内容記述	この博士論文は内容の要約のみの公開（または一部非公開）になっています
発行年	2018
学位授与大学	筑波大学 (University of Tsukuba)
学位授与年度	2017
報告番号	12102甲第8528号
URL	http://hdl.handle.net/2241/00152400

博士（工学）論文要約

ショートリードマッピング
高速計算システムの研究

システム情報工学研究科　知能機能システム専攻

曽我部　陽光

２０１８年3月

本研究は，書き換え可能なLSIであるField Programmable Gate Array（以下，FPGAと呼ぶ）と画像処理用のプロセッサであるGraphics Processing Unit（以下，GPUと呼ぶ）を用いて，遺伝子情報処理分野におけるショートリードマッピングと呼ばれる問題を高速に処理する計算機システムの構築を行う．

ショートリードマッピングは，ある個体のDNA配列（アデニン（A），シトシン（C），グアニン（G），チミン（T）の4種類の塩基の並び）を解読する（この処理は，DNAシーケンシングと呼ばれる）ために必要な処理であり，医療，創薬，農業，生態学など幅広い応用範囲を持つ．以下に，DNAシーケンシングの流れを説明し，ショートリードマッピングの位置づけを述べる．DNAの塩基配列を高速に読み取る装置を，DNAシーケンサーと呼び，この装置から得られるデータは短いDNA断片（ショートリードと呼ぶ）である．DNAシーケンサーによって読み出されたショートリードは，全体のDNA配列のどこに位置するかは不明であり，このままでは遺伝子解析に利用できない．モデル生物の場合は既にゲノムが解読されているため，各ショートリードについて，その生物種の既知のゲノム（参照配列と呼ぶ）中で対応する位置を見つけることで，そのショートリードが全体のDNA配列のどこに位置しているかを特定する（この処理をショートリードマッピングと呼ぶ）．最後に，マッピングされたショートリードと参照配列の比較から読み出されたDNAの遺伝的な特徴（参照配列に対する変異）を得る．このように，ショートリードマッピングは，DNAシーケンシングに必要な処理であり，ショートリードマッピングの結果を通して，解析対象のDNAの遺伝的特徴が得られる．

ショートリードマッピングには，技術的な課題が主に2つある．第一に，処理するショートリード数が膨大なため，処理速度が重要である．例えば，ヒトのDNA解析の場合は約10億個のショートリードを処理することが必要となる．DNAシーケンサーは，非常に高速であるため，ショートリードマッピングがDNAシーケンシングのボトルネックとなっており，その高速化が課題となっている．また，DNAシーケンサーは，性能および低価格化がムーアの法則を超える速度で向上を続けており，ショートリードマッピングの高速化の需要は，ますます増大すると言われている．第二に，ショートリードと参照配列が100%一致するとは限らず，ある程度変異を含んでいてもうまくマッピングできることが求められる．これを踏まえ，本研究では，処理速度とマッピング精度の2つの観点から評価する．

ヒトのDNAシーケンシングについては，オーダーメイド医療と呼ばれる，個人のDNA配列に基づいた医療の実現が期待されているため特に需要が高く，本研究では対象をヒトに限定して議論するが，これは議論の複雑化を避けるためであり，他の生物種の場合も適応可能である．オーダーメイド医療実現のためには，DNAシーケンシングの高速化・低コスト化が重要であると言われており，具体的には，一人のDNAシーケンシングに1時間・1万円という目標が挙げられている．要求される性能を実現するために，マッピング処理に大規模な計算機システムを利用すれば，DNAシーケンシングのコストが増加してしまう．そこで，個人や大学の研究室レベルで導入可能な小規模なコンピュータシステムで高性能な処理を実現することが重要であると考え，本研究では，書き換え可能なLSIであるFPGAと画像処理用の演算ユニットであるGPUを用い，FPGAまたはGPUを付加したパーソナルコンピュータ（以下，Host-PCと呼ぶ）という比較的小規模な計算機システムで，ショートリードマッピングの高速化を図る．

FPGA，GPU，いずれの場合に置いても，高速化を実現する上で重要なのは，並列処理である．ショートリードマッピングでは，ショートリード毎，あるいはseed毎に独立して処理することが可能であり，容易に並列処理することができる．ヒトの全ゲノムシーケンシングを行う場合，100塩基長のショートリードを10億本程度処理する必要があるが，このショートリードは全て並列に処理可能であり，潜在的な並列性は非常に高い．一方で，ショートリードリードから抜き出したseedの参照配列中での出現位置を高速に検索するために，事前にhash tableを作成しておき，そのhash tableを読み込んで検索するが，そのサイズは数GB以上になるためオフチップ・メモリに保存されており，処理ごとにプロセッサに読み込んでくる必要がある．プロセッサの並列演算能力に比べ，メモリアクセスは遅く，処理のボトルネックになる．ショートリードのデータ次第でアクセス先も変わるため予めオンチップ・メモリに読み込んでおくことはできず，メモリアクセス局所性が低いため，キャッシュするという方策は有効に働かない．このような理由から，ソフトウェアで用いられるアルゴリズムをそのまま適応しても，メモリアクセスがボトルネックになり，大規模な高速化は難しい．本研究では，メモリアクセスがボトルネックとならずFPGA及びGPUの回路資源量に応じて高速化可能なソートと並列比較に基づくショートリードマッピング並列処理アルゴリズを提案する．本手法は，大きく分けて以下の5種類の要素から構成される:(1) ソー

トと並列比較，(2)segment indexing，(3)flexible match，(4) 可変 mask を用いた hash，(5) 解候補の確率的削除.

(1) ソートと並列比較は，提案手法の最も根幹の要素であり，複数のデータに対して同時にメモリ参照することで，メモリアクセス・ボトルネックを回避することが目的である．初めにショートリードから抜き出した seed を hash 値でソーティングして同種のデータを集め，集まった seed を 1000 個以上同時に処理する．これにより，オフチップ・メモリに格納された hash table へのアクセス回数は，1/1000 以下になり，メモリアクセスのボトルネックを回避できる．ただし，高速にソートするためにバケットソートを用いるが，これが hash 値の定義域を狭め，同一 hash bucket に格納される要素数を大幅に増加させる．同一の hash bucket に格納されたものから，目的の要素を見つけ出すための計算量は増加（平均比較回数の増加）するが，この処理は，1000 seed 以上並列に行い，また後述する (2) の手法が総要素数の削減を行うため，全体としては高速化できる．バケットソートについては，FPGA システムでは FPGA 上で，GPU システムでは，ソフトウェア（host-PC 上）で行う．FPGA では，オンチップ・メモリを用いて，各 seed をキャッシュし，複数の seed を連続してオフチップ・メモリに書き出すことで，メモリ転送の高効率化を図る．GPU では，FPGA のようにオンチップ・メモリを自由に構成することが不可能のため，同様の手法は困難であり，Host-PC 上で処理する．このとき，Host-PC と GPU は非同期的に動作し，Host-PC がソート処理を行っている間も，GPU は別の処理を行うため，Host-PC はソート処理の時間は全体の処理時間にほとんど影響を与えないように設計した.

(2)segment indexing は，参照配列を一定距離で segment に分け，その中で 1 つの seed だけを hash table に登録する．segment indexing では，segment の長さ s とすると，hash table の総要素数は，従来の $1/s$ になり，hash bucket 内の比較回数を減らすことができる．また，従来手法で，出現回数が 8 回より多い seed を除外していたのを条件付きで緩和することで，マッピング精度の向上を図る．この手法は，データ転送量・比較回数を削減するため高速化に寄与するが，マッピング精度低下を行き起こす．そのため，後述する (3)flexible match を用いることで，マッピング精度の向上を図る.

(3)flexible match は，マッピング精度の向上を目指すもので，ショートリードから抜き出した seed と，参照配列から抜き出し hash table に登録された seed を比較する際に，1 塩基までの変化を許容する比較（以

下，flexible match と呼ぶ）を行う．flexible match は，FPGA では回路資源の投入，GPU では複数のビット演算の組み合わせで実現される．この flexible match は，比較演算が複雑になるため処理速度には悪影響を与えるが，seed 中の変異を打ち消せるためマッピング精度の向上が期待できる．(2)，(3) を組み合わせて考えた場合，従来と同等以上のマッピング精度で大幅な高速化を実現できることを実験的に示した．

(4) 可変 mask を用いた hash は，hash 関数そのものの改善を目的とする．DNA 配列は，特定の並びが反復して出現することが多いため，hash の衝突が不均一になりやすい．そこで，事前に参照配列から mask の表を計算しておき，この表を元に反復して出現する DNA 配列の並びには大きな mask を用いて細かく hash，逆にほとんど出現しない DNA 配列の並びには小さな mask を用いて荒く hash することにより，全体として，均一に hash できるようにする．本手法は，hash の総衝突数（平均比較回数）を半分以上に削減できることを実験により示した．

(5) 解候補の確率的削除とは，ショートリードから抜き出した seed から，hash table を用いて解候補を見つけたときに，その解候補について Smith-Waterman 法（SW 法）を用いてスコアを計算するかを確率的に決定する手法である．SW 法を計算する回数を減らすことが目的である．スコアの理論的な上限値と暫定的なスコアの差を元に，新しく得られた CAL を SW 法を用いてスコア付けを行うかどうか，を確率的に決定する．この方法は，SW 法を適応する数を減らすことができる一方で，高いスコアの解候補を捨ててしまう可能があるため，マッピング精度を低下させる可能性がある．実験によると，SW 法の総計算回数を半分にするときに，マッピング精度の低下は 0.01%以下であったため，本手法は有効であることが示された．

本研究は，新たに並列処理手法を提案し，FPGA または GPU を PCI Express で接続したパーソナルコンピュータいう小規模なショートリードマッピング高速計算システムを構築した．本手法の有効性を検証するための性能評価を行った．まず，ショートリードと参照配列間の変位量とマッピング精度の関係を正確に測定するために，参照配列から人為的に変異を加えたショートリードを模擬データとして作成し，他の手法とマッピング精度の比較を行った．その結果，最もマッピング精度の高いソフトウェアの一つである BWA と比べて，本手法は，変異が少ない場合で同等，変異が多い場合で優位な結果を示した．次に，DNA シーケンサーが生成した実データを用いて処理速度を測定を行った．最有力なソフトウェ

アマッピングツール BWA と比較して，FPGA システムで 18 倍，GPU システムで 8 倍程度の高速化を達成した．この処理速度は，ヒト一人分の DNA の塩基配列解読をそれぞれ 1 時間程度，2 時間程度で処理可能な性能である．本研究は，FPGA と GPU のどちらがショートリードマッピングに向いているか，効率的に処理できるかを明らかにすることを目的にしているものではない．FPGA と GPU は，価格，消費電力，柔軟性などが異なるため，状況に応じて最適な選択が異なり，両者は並立できるものだと考えている．本研究は，FPGA システム，GPU システム，双方において高い性能を示したことが重要であり，その成果は低価格で高速な DNA 解析の実現に貢献するものである．

また，本研究で提案した手法は，従来のハードウェアシステムとは異なり，メモリ転送速度の制約をほとんど受けずに，ハードウェアの規模に応じて性能向上可能である．これは，提案手法が，本論文で示したハードウェア構成において他の手法に対して優れているだけでなく，将来開発されるであろうより大規模なハードウェアを用いれば，より優れた性能が実現できることを意味する．将来の DNA シーケンサーの性能（ショートリードの生成速度）向上，DNA シーケンシングの需要拡大が予想されているが，CPU の性能向上が鈍化しているため，CPU の性能に依存するソフトウェアアプローチでは，将来求められる性能に対応できない．提案手法を用いれば，より大規模なハードウェアを用いることで将来においても十分な性能が達成できると考えられる．

最後に，本研究の情報科学分野一般における貢献について述べる．ショートリードマッピングは，処理の潜在的な並列性が非常に高い一方で，大規模なデータに対して局所性が低い参照を頻繁に行う問題である．このような問題は，メモリ転送速度がボトルネックとなるため，FPGA や GPU を用いた並列処理で大幅な高速化を実現することはできないと考えられて来た．本研究は，その定説に挑戦したものであり，本研究の成果を通して，他の同様の問題に対しても，突破口のヒントを与えうるものである．